



TECHNICKÁ UNIVERZITA V LIBERCI
Fakulta mechatroniky, informatiky
a mezioborových studií



REALIZACE PROGRAMOVÉHO MODULU PRO IDENTIFIKACI SOUSTAVY ODPOROVÉHO VÝPARNÍKU

Bakalářská práce

Studijní program: B2612 – Elektrotechnika a informatika
Studijní obor: 2612R011 – Elektronické informační a řídicí systémy
Autor práce: **Tomáš Šimůnek**
Vedoucí práce: Ing. David Vápenka





TECHNICAL UNIVERSITY OF LIBEREC
Faculty of Mechatronics, Informatics
and Interdisciplinary Studies ■

DEVELOPMENT OF PROGRAM MODULE FOR IDENTIFICATION OF A THERMAL EVAPORATOR SYSTEM

Bachelor thesis

Study programme: B2612 – Electrical Engineering and Informatics
Study branch: 2612R011 – Electronic Information and Control Systems
Author: **Tomáš Šimůnek**
Supervisor: Ing. David Vápenka



ZADÁNÍ BAKALÁŘSKÉ PRÁCE

(PROJEKTU, UMĚLECKÉHO DÍLA, UMĚLECKÉHO VÝKONU)

Jméno a příjmení: **Tomáš Šimůnek**
Osobní číslo: **M11000177**
Studijní program: **B2612 Elektrotechnika a informatika**
Studijní obor: **Elektronické informační a řídicí systémy**
Název tématu: **Realizace programového modulu pro identifikaci soustavy odporového výparníku**
Zadávající katedra: **Ústav mechatroniky a technické informatiky**

Z á s a d y p r o v y p r a c o v á n í :

1. Prozkoumejte současný stav řízení rychlosti napařování u vakuové napařovačky a sestavte přehled funkcí, které musí programový modul zajišťovat.
2. Nastudujte metody experimentální identifikace dynamických soustav a možnosti implementace těchto metod ve vývojovém prostředí ControlWeb.
3. V prostředí ControlWeb naprogramujte modul splňující požadavky vzniklé v bodě 1 tohoto zadání.
4. Proveďte několik identifikačních cyklů a zhodnoťte funkčnost a spolehlivost identifikace a kvality regulace PID regulátoru, nastaveného na základě identifikovaného modelu systému.

Rozsah grafických prací: **dle potřeby dokumentace**
Rozsah pracovní zprávy: **30–40 stran**
Forma zpracování bakalářské práce: **tištěná/elektronická**
Seznam odborné literatury:


- [1] Havlena V., Štecha J.: Moderní teorie řízení. 2. vydání. Skripta ČVUT, Fak. elektrotechnická, 2000
- [2] Soukup J.: Identifikace soustav. 1. vydání. Skripta ČVUT, Fak. strojní, 2000

Vedoucí bakalářské práce: **Ing. David Vápenka**
Ústav mechatroniky a technické informatiky
Konzultant bakalářské práce: **Ing. Leoš Kukačka**
Ústav mechatroniky a technické informatiky

Datum zadání bakalářské práce: **10. října 2013**
Termín odevzdání bakalářské práce: **16. května 2014**


prof. Ing. Václav Kopecký, CSc.
děkan

L.S.


doc. Ing. Milan Kolář, CSc.
vedoucí ústavu

V Liberci dne 10. října 2013

Prohlášení

Byl jsem seznámen s tím, že na mou bakalářskou práci se plně vztahuje zákon č. 121/2000 Sb., o právu autorském, zejména § 60 – školní dílo.

Beru na vědomí, že Technická univerzita v Liberci (TUL) nezasahuje do mých autorských práv užitím mé bakalářské práce pro vnitřní potřebu TUL.

Užiji-li bakalářskou práci nebo poskytnu-li licenci k jejímu využití, jsem si vědom povinnosti informovat o této skutečnosti TUL; v tomto případě má TUL právo ode mne požadovat úhradu nákladů, které vynaložila na vytvoření díla, až do jejich skutečné výše.

Bakalářskou práci jsem vypracoval samostatně s použitím uvedené literatury a na základě konzultací s vedoucím mé bakalářské práce a konzultantem.

Současně čestně prohlašuji, že tištěná verze práce se shoduje s elektronickou verzí, vloženou do IS STAG.

Datum:

Podpis:

Poděkování

Chtěl bych poděkovat ing. Davidu Vápenkovi za odborné vedení, trpělivost a cenné rady při tvorbě této bakalářské práce.

Abstrakt

Bakalářská práce se zabývá identifikací odporového výparníku aparatury pro napařování tenkých vrstev. Cílem této práce je zautomatizování procesu kvůli zvýšení efektivity obsluhy napařovačky a umožnění jejího ovládání bez odborné znalosti. V práci jsou pro identifikaci odporového výparníku použity metoda aproximace prvním řádem a metoda nejmenších čtverců. Pro identifikaci byl naprogramován modul ve vývojovém prostředí Control Web. Data se povedla úspěšně identifikovat oběma metodami, nemohla však být realizována na reálné soustavě. Identifikace byla simulována na modelovém ovladači systému Control Web a zároveň jsme pomocí modulu analyzovali již naměřená data. Výsledky této práce umožňují zkvalitnění a zefektivnění práce napařovací aparatury, přičemž se bude muset funkčnost modulu ověřit v reálných podmínkách.

Klíčová slova

Control Web, metoda nejmenších čtverců, aproximace prvním řádem, identifikace, napařovací aparatura

Abstract

The bachelor thesis focuses on the identification of resistance vaporizer apparatus for the deposition of thin layers. The aim of this thesis is to automate the process to increase operator efficiency steaming equipment. We use to identify the resistance vaporizer first order approximation method and the least squares method. To identify the module was programmed in the development environment Control Web. Data was successful identified by both methods but could not be implemented on the real system. Identification was simulated on the model driver of system Control Web and we also analyzed using the existing data. The result of this work enables a better and more efficient work of steam apparatus while it will have to verify the functionality on the real system.

Key Words

Control Web, the least squares method, first order approximation, identification, steaming equipment

Obsah

Úvod	11
1. Vakuová napařovací aparatura VNA 600/2.....	12
1.1 Popis.....	13
1.1.1 Schéma ovládání	13
1.1.2 Recipient aparatury	15
1.2 Funkce napařovací aparatury	16
2. Vývojové prostředí Control Web	17
2.1 Obecný popis.....	17
2.1.1 Datové elementy	18
2.1.2 Datové typy.....	18
2.1.3 Dvojcestné programování	19
2.1.4 Časování aplikace	20
2.1.5 Ovladače	21
2.2 Přístroje	21
2.2.1 Parametry a procedury	21
2.2.2 Příklady přístrojů	22
2.3 Simulace v systému Control Web	23
2.3.1 Modelový ovladač.....	23
2.3.2 Nástroj pro ladění aplikace	24
3. Identifikace dynamických systémů	25
3.1 Identifikace.....	25
3.1.1 Matematicko-fyzikální metody identifikace	26
3.1.2 Experimentální metody identifikace	27
3.2 Struktura lineárních modelů	28
3.2.1 Lineární model ve spojitém čase	28
3.2.2 Tvary lineárních modelů v diskrétním čase	28
3.3 Vybrané metody identifikace	31
3.3.1 Aproximace 1. řádem s dopravním zpožděním	31
3.3.2 Metoda nejmenších čtverců	32

4. Návrh konstant PID regulátoru	36
4.1 Převod ze Z obrazu do Laplaceova obrazu	36
4.2 Přímá syntéza regulátoru.....	37
4.3 Výrobce zadane vztahy pro určení konstant PID regulátoru	39
5. Modul pro identifikaci.....	40
5.1 Popis modulu.....	40
5.2 Proces identifikace	41
5.3 Metody identifikace	42
6. Zhodnocení výsledků	44

Seznam zkratek

ARMAX	Autoregressive moving average model – druh lineárního diskrétního model
ARX	AutoRegresive model with eXternal input – druh lineárního diskrétního model
BJ	Box-Jenkins – druh lineárního diskrétního model
CCD	Charge-coupled device – elektronická součástka
ITAE	Integral of Time Absolute Error – integrální kritérium
LCD	Liquid Crystal Display – display s tekutými krystaly
MS Windows XP	Microsoft Windows XP – operační systém
OCL	Object Control Language – jazyk řízených objektů
OE	Output Error – druh lineárního diskrétního model
PCI	Peripheral Component Interconnect – počítačová sběrnice
PID regulátor	Proporcionálně integračně derivační regulátor
SISO	Single Input Single Output – model jednoho vstupu, jednoho výstupu

Úvod

Z důvodu potřeby člověka řídit a regulovat veškeré procesy výroby je také nutná znalost charakteru systému či dokonce jeho matematického modelu. Tato znalost nám dává možnost správného nastavení řídicího pochodu. Tato bakalářské práce se zaměřuje na identifikaci soustavy vakuové napařovačky, konkrétně rychlosti napařování materiálu na substrát. K tomuto účelu je naprogramován modul pro identifikaci systému.

Snahou naprogramovaného modulu pro identifikaci odporového výparníku je zautomatizování procesu napařování tenkých vrstev při zachování či zlepšení vlastností výsledného pochodu. Důvodem automatizace je především zvýšení efektivity obsluhy napařovačky a dále zajištění, aby obsluha nemusela mít odbornou znalost z partií identifikace a návrhu parametrů regulátoru.

Programový modul je realizován ve vývojovém prostředí Control Web. V tomto prostředí je programován celý ovládací systém napařovací aparatury. Práce seznamuje s tímto vývojovým prostředím a možnostmi implementace experimentálních metod identifikace. Jako experimentální metody pro identifikování soustavy odporového výparníku byly vybrány aproximace prvním řádem se zpožděním a dále komplexnější metoda nejmenších čtverců se zpožděním. Bude popsána přímá syntéza PID regulátoru sloužící k určení jeho parametrů pro zajištění správných vlastností procesu. Nakonec jsou zhodnoceny výsledky identifikace a dále kvalita regulace PID regulátoru nastaveného na základě zjištěného modelu.

Napařovací aparatura se nachází v centru *TOPTEC*, regionálním centru speciální optiky a optoelektronických systémů, spadající pod *Ústav fyziky plazmatu AVČR*.

1. Vakuová napařovací aparatura VNA 600/2

V této části práce je popsána napařovací aparatura VNA 600/2, především pak partie týkající se regulace proudu odporovým výparníkem. V první podkapitole popisujeme samotnou aparaturu a zjednodušené schéma toho, jakým způsobem je zapojené ovládání. V další části se věnujeme funkcím, ke kterým je soustava určena.



Obrázek 1: Napařovací aparatura; obrázek převzat z dokumentace VNA 600/2¹

¹ PZP. VNA 600/2: Vakuová napařovací aparatura. Dobruška, 2004.

1.1 Popis

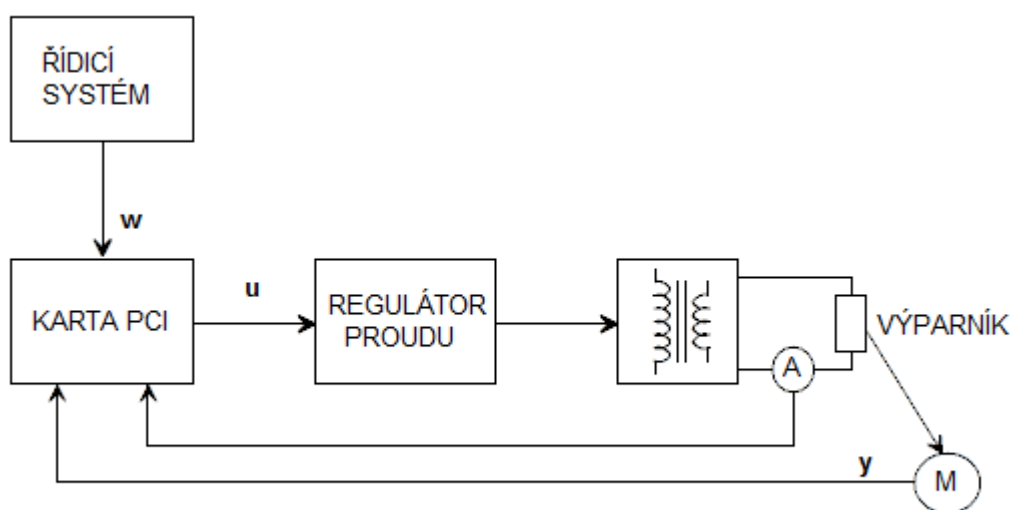
Sestava VNA 600/2 se skládá ze dvou částí. První je recipient s čerpací aparaturou tvořenou z rotační olejové a turbomolekulární vývěvy. Druhou částí je skříň s elektroinstalací, regulátorem a řídicím systémem.

1.1.1 Schéma ovládání

V níže uvedeném schématu (obr. 2) je znázorněno zapojení regulátoru pro ovládání průchodu proudu odporovým výparníkem. V řídicím systému, který zde zastupuje počítač s programem pro ovládání v prostředí Control Web, je nastavena požadovaná hodnota rychlosti napařování w . Tato hodnota je poté zpracována kartou PCI s PID regulátorem, který řídí tyristorový regulátor proudu hodnotou řídicího napětí 0-10 V. Výstupní proud je v regulačním obvodu tedy akční veličinou u . Za regulátorem je zapojen transformátor měnící napětí s transformačním poměrem 100:1, kde na sekundární vinutí je připojen odporový výparník, jenž nám představuje dohromady s odpařujícím se materiálem systém, který je regulován. Za odporovým výparníkem je zapojen proudový transformátor, který zpět přivádí hodnotu proudu do řídicího systému. Dále je měřena rychlost napařování pomocí krystalového senzoru umístěného nad výparníkem. Rychlost napařování představuje výstupní hodnotu regulačního procesu a uzavírá zpětnovazební obvod, protože informace o rychlosti je přivedena zpět do regulátoru na kartě PCI. Všechny hodnoty přivedené zpět na vstup karty lze v řídicím systému načítat a analyzovat.

Řídicím systémem je zde počítač platformy PC ovládaný pomocí dotykového LCD, pracuje pod operačním systémem MS Windows XP. Konkrétní aplikace pro řízení je vytvořena v prostředí Control Web. Pro komunikaci s aparaturou je použita speciální PCI karta pro nanašení tenkých vrstev SQM242. Výrobce karty firma INFICON v dokumentaci uvádí možnost přepínání manuálního či automatického módu regulace.

Automatický režim zajišťuje regulátor PID². Tyristorový regulátor je od firmy *ELFIS* typu *ELFIS*tor 1FI25/400. Podle dokumentace výrobce jeho označení znamená: 1F – jednofázové provedení, I – regulátor proudu, 25 – jmenovitý proud v ampérech, 400 – jmenovité napětí ve voltech³. Transformátor má na sekundárním vinutí parametry 4 V/1000 A. Krystalový měřicí senzor rychlosti napařování pracuje na frekvenčním rozsahu 1 MHz – 10 MHz a kartou je následně zpracovávána hodnota rychlosti napařování. Rychlost je dále integrována na tloušťku již naneseného materiálu.



Obrázek 2: Schéma regulačního pochodu napařování

Pro zjednodušení je na obrázku 2 zobrazen pouze jeden transformátor a jeden výparník. Reálná soustava má transformátory a odporové výparníky dva. Mezi nimi přepíná stykač, který má navíc ještě 3. polohu, kdy není sepnut ani jeden výparník.

² SQM-242 Thin Film Deposition Controller Card. INFICON [online]. 2013 [cit. 2014-05-05]. Dostupné z: <http://products.inficon.com/en-us/nav-products/Product/Detail/SQM-242?path=Products%2FThinFilmDeposition%2F>

³ Tyristorové regulační jednotky ELFISstor 1F. ELFIS [online]. [cit. 2014-05-04]. Dostupné z: http://www.elfis.cz/website/download/pdf_elfistory/1fx/_downloadFile.php

1.1.2 Recipient aparatury

Recipient je součást aparatury, ve které je nanášen materiál na substrát. Vnitřní prostor komory (obr. 3) je pokryt hliníkovou fólií, aby bylo zjednodušeno čištění potřebných partií. V recipientu se nachází několik důležitých částí, které níže popíšeme. Na obrázku č. 3 jsou ukázány.

- 1) *Halogenová žárovka* pro zahřívání cílového předmětu proto, aby se materiál lépe nanášel.
- 2) *Držák na rotační planetární přípravek* na uchycení předmětu, na který je materiál nanášen. Tento přípravek zajišťuje rovnoměrné rozprostření nanášené vrstvy.
- 3) *Frekvenční senzor* rychlosti napařování.
- 4) *Clonky* pro zadržení odpařovaného materiálu, používají se, než se materiál dostatečně prohřeje.
- 5) *Dva odporové výparníky* připojené do měděných elektrod, které jsou zespodu chlazeny vodou, aby nedocházelo k jejich odpařování.

Dále je v recipientu umístěno čidlo pro kontrolu teploty.



Obrázek 3: Vnitřní část recipientu

1.2 Funkce napařovací aparatury

Napařovací aparaturu lze používat jak pro produkční, tak pro laboratorní napařování tenkých vrstev. Cílem napařovacího procesu je nanést několik vrstev odpařovaného materiálu na substrát.

Předměty, na které lze nanášet tenké vrstvy, mohou být jak kovové, tak nekovové. Nejčastěji používaným materiálem je však sklo, kde upravujeme jeho optické vlastnosti, abychom dosáhli požadované spektrální charakteristiky. Nanášenými materiály jsou například kovy jako hliník, chrom, zlato, stříbro, měď, nikl a různé sloučeniny kovu jako fluorid hořečnatý, sulfid zinečnatý i nekovové jako je oxid křemičitý. Výparníky jsou drátky nebo do tvaru lodiček či mističek natvarované různé kovy s dostatečně vysokým bodem tání, aby se samy co nejméně odpařovaly při zahřívání a zároveň nereagovaly s odpařovaným materiálem. Nejvhodnějšími kovy pro tuto funkci je wolfram, který má bod tání nejvyšší ze všech kovů, tantal, molybden. Případně lze použít i mističku z keramiky, například oxidu hlinitého, omotanou wolframovým drátkem.

Podle potřebných optických vlastností se nanáší různý počet vrstev o různé tloušťce. Například pro jednoduchou antireflexní vrstvu se na skleněnou čočku nanáší 1 vrstva fluoridu hořečnatého o tloušťce od 100 nm do 300 nm. Obecně se nejčastěji nanáší vrstvy od 60 nm do 1 μ m. Doba napařování je úměrná potřebné tloušťce vrstvy a rychlosti napařování. Například pro napaření 100 nm vrstvy je potřeba nejdříve postupné zvyšování výkonu pro předtavení materiálu po dobu cca 1 minuty při zavřené clonce. Poté se při průměrné rychlosti 0.4 nm/s přibližně 4 minuty napařuje a nakonec je potřeba opět po rampě výkon snížit - přibližně 20 s. Celkově napařování jedné vrstvy o 100 nm trvá tedy cca 6 minut.

Před každým započatým procesem napařování se musí nejprve komora odvzdušnit na žádaný tlak, k čemuž jsou určeny vývěvy. Doba odčerpávání se odvíjí od žádaného podtlaku a použité vývěvy, vždy se však jedná o hodinu až hodinu a půl.

2. Vývojové prostředí Control Web

V této kapitole se budeme zabývat vývojovým prostředím Control Web verze 6. Vývojové prostředí Control Web je od firmy *Moravské přístroje a.s.* Jedná se o univerzální nástroj pro rychlý vývoj aplikací s objektově orientovanou komponentovou architekturou. Sám výrobce uvádí v dokumentaci systému: „*Hlavním cílem návrhu systému Control Web je učinit realizaci běžných úkolů snadnou a neobvyklých, komplikovaných úkolů možnou.*“⁴

Snahou výrobců bylo tedy vytvořit velice univerzální nástroj s širokým uplatněním jak v řízení velkých procesů, přímém řízení strojů, tak i v možnostech vizualizace právě probíhajících pochodů. Control Web 6 je vzestupně kompatibilní s předchozími verzemi, tudíž přechod mezi nimi by neměl znamenat problém.

Firma *Moravské přístroje a.s.* se ovšem nezabývá pouze systémem Control Web. Sama nabízí velice širokou sadu řešení pro průmyslové aplikace. Mezi produkty patří dále programový systém *VisionLab* pro strojové vidění, který může mít uplatnění např. při různém strojovém měření vzdáleností, rozměrů, ale i pro rozpoznávání tvarů. K tomu patří i komponenty jako jsou *CCD kamery* pro snímání obrazu a osvětlovací jednotky. Dále firma vyrábí vlastní počítačový systém *DataLab*, do kterého spadají různé průmyslové jednotky pro připojení vstupů a výstupů, průmyslové dotykové LCD a dokonce i počítače platformy PC pro náročné aplikace. Celkově tak výrobce tvoří komplexní řešení pro mnoho aplikací.

2.1 Obecný popis

Pomocí vývojového prostředí Control Web lze velice rychle a efektivně vytvořit aplikace okamžitého použití. Projekty se ukládají do souboru formátu s příponou *cw*. Po založení nového projektu se zobrazí prázdná plocha obsahující pouze přístroj Panel, na který se nejčastěji vkládají ostatní přístroje. Jak už jsme výše zmínili, Control Web

⁴ *Control Web 6 – dokumentace. Moravské přístroje a.s.* [online]. 2013 [cit. 2014-05-04]. Dostupné z: <http://www.mii.cz/cat?id=98&lang=405>

má objektově orientovanou komponentovou architekturu. Základními kameny tvorby jsou tedy objekty – přístroje. Programování probíhá pomocí jazyka řízení objektů známého jako *OCL*.

2.1.1 Datové elementy

Každý program potřebuje ke své realizaci data, která bývají rozdělována podle druhů. Control Web rozeznává několik druhů datových elementů, podle toho, k čemu jsou určeny. Datovými elementy jsou:

- *Konstanta* – má určenou hodnotu a ta se během programu nemění.
- *Proměnná* – ta je pak dále určena, jestli je skalární (jedna hodnota), pole (seznam hodnot) nebo buffer (blok dat).
- *Kanál* – určen pro spolupráci s ovladači, může být vstupní, výstupní nebo obousměrný.
- *Výrazy* – určeny k úpravě vstupních dat, používají se pro přímý přepočítání vstupních hodnot, v programu se vyčíslují při jejich volání.
- *Schedule* – stejné jako výrazy, ale jsou vyhodnocovány periodicky.

U každého datového elementu musí být určen datový typ, respektive jakým formátem dat je určen.

2.1.2 Datové typy

Vývojové prostředí rozeznává datové typy, které vidíme v tabulce 1.

typ	identifikátor	délka [byte]	rozsah	poznámka
logický	boolean	1	false..true	binárně 0..1
číselný	shortint	1	-128 až 127	
	shortcard	1	0 až 255	
	integer	2	-32 768 až 32 767	
	cardinal	2	0 až 65 535	
	longint	4	-2 147 483 648 až 2 147 483 647	
	longcard	4	0 až 4 294 967 295	
	shortreal	4	+/- 1,2× 10 ⁻³⁸ až 3,4× 10 ³⁸	IEEE 754 32bit
	real	8	+/- 2,3× 10 ⁻³⁰⁸ až 1,7× 10 ³⁰⁸	IEEE 754 64bit
řetězcový	string	různá	—	nesmí obsahovat znak „0”
datový	data	různá	—	
blokový	buffer	min. 256	—	velikost se určuje po blocích

Tabulka 1: Datové typy prostředí Control Web⁵

Z tabulky 1 popisuje datové typy, které rozeznává Control Web. Mezi klasické datové typy patří 1 logický, 8 číselných datových typů a jeden řetězcový. Dále zde je nestandardní datový typ pro obecná nspecifikovaná data. Tento typ pouze zabírá paměť, ale ani sám Control Web nerozpoznává, jakého formátu v něm jsou data uložena. Proto jsou velice omezené možnosti práce s ním. Lze ho pouze přiřazovat, posílat jako parametr do procedury a porovnávat rovnost či nerovnost se stejným datovým typem. Dalším nestandardním typem je datový typ pro bloková data. Tento typ určuje svou vnitřní strukturu podle toho, k čemu je používán. Lze do něj například vložit blok jednoduchých datových typů.

2.1.3 Dvojcestné programování

Dvojcestné programování znamená vytvářet aplikaci z části graficky a z části textově. Důvod k této technice je prostý. Sestavení komponent, ze kterých se bude aplikace skládat, je mnohem jednodušší pomocí grafického programování, ovšem některé funkčnosti je potřeba dodefinovat pomocí textového programování.

⁵ *Control Web 6 – dokumentace. Moravské přístroje a.s.* [online]. 2013 [cit. 2014-05-04]. Dostupné z: <http://www.mii.cz/cat?id=98&lang=405>

V reálném programování to vypadá tak, že se překlápí mezi textovým režimem se zdrojovým kódem a plochou sestavení aplikace (grafický režim). Při přechodu z grafického režimu je potom provedeno generování kódu a při přechodu z textového režimu do grafického je proveden předklad kódu.

Grafické programování probíhá především pouze za pomoci myši počítače, kdy přetahujeme z palety s přístroji na pracovní plochu přístroje, které potřebujeme a měníme jejich vlastnosti jako je vzhled atp.

Textově lze naprogramovat celou aplikaci, ovšem bylo by to na úkor rychlosti. Nejčastěji se proto přímo programuje pouze v procedurách přístrojů.

Jazyk programování obsahuje všechny základní funkčnosti a lze s ním jednoduše algoritmizovat. Má velice intuitivní syntaxi podobnou například jazyku *Pascal*.

2.1.4 Časování aplikace

Každá aplikace potřebuje nějaký zdroj své aktivity. V Control Webu lze mít aplikaci řízenou buď daty (spíše vizualizační účely) nebo časově (pro oblast řízení).

Zásadní rozdíl v těch dvou možnostech je v aktivitě přístrojů. V časově řízených aplikacích jsou přístroje aktivovány nejčastěji periodicky, případně vyvolanou výjimkou od přístroje či ovladače. V datově řízených aplikacích většina aktivací probíhá v reakci na změnu dat.

Časově řízené aplikace jsou často také nazvány aplikace reálného času. Systém Control Web je časován operačním systémem a jeho časové rozlišení se odvíjí od rozlišení operačního systému. V systémech Windows XP a Windows 2000 jde o rozlišení cca 10 ms. Aplikaci lze navíc časovat relativně nebo absolutně. Rozdíl těchto možností je v tom, že relativní časování se odvíjí od startu aplikace, tedy že perioda aktivací přístrojů se neodvíjí od nějakého jasně daného časového datu. Na rozdíl od absolutního časování, kde se perioda spouští od jasně daného časového údaje s možností posunutí první periody. Například od půlnoci a posunutím 30 sekund a periodou jedné minuty.

2.1.5 Ovladače

Pro komunikaci s okolními zařízeními jsou v Control Webu určeny ovladače. Ovladač je vložen do aplikace v záložce *Ovladače*. Každému ovladači musí být navíc přiřazen jeho mapovací a parametrický soubor. Pro čtení a zápis hodnot ovladače se musí vytvořit kanály.

Mapovací soubor systému říká, jaké jsou dostupné kanály ovladače, jejich typ a směr. Tyto informace slouží pro kontrolu překladači. Tento soubor bývá zpravidla dodán s ovladačem.

Parametrický soubor obsahuje skutečné parametry ovladače a jeho komunikačních kanálů. Tento soubor obsahuje specifická data pro komunikaci s ovladačem. Na rozdíl od mapovacího souboru (který slouží pouze Control Webu), tento soubor slouží pro nastavení skutečného ovladače.

2.2 Přístroje

Přístroje je možné přidávat pomocí záložky *Paleta přístrojů*. V této záložce jsou přístroje různě členěny do kategorií a podkategorií, podle toho, jakou funkčnost zastávají. Přístroje mohou být přidány jako viditelné nebo neviditelné. Viditelné přístroje lze vidět přímo na ploše *Panelu*, kde se s nimi může při běhu aplikace viditelně pracovat (vypínat, psát hodnoty atp.), neviditelné přístroje jsou přidány pouze pro svou funkčnost, nelze s nimi přímo pracovat při běhu aplikace, ale lze například vyvolávat jejich procedury.

2.2.1 Parametry a procedury

Nastavení přístroje se děje pomocí dialogového okna *Inspektor přístroje*. Všechny vlastnosti jsou vyjmenovány v záložce *Parametry*. Přístroji lze navíc definovat lokální proměnné, které jsou vidět pouze uvnitř přístroje. Často ovšem nestačí jen nastavit parametry, protože potřebujeme určitou funkčnost přístroje, kterou ale samotné nastavení parametrů nenabízí. K tomu jsou zde procedury přístroje.

Každý přístroj má své 3 druhy procedur:

- 1) *Událostní procedury* – tyto procedury jsou řízeny událostmi, které se s přístroji dějí. Příkladem může být procedura *OnPress()* přístroje *Button*, která se vyvolá při jeho zmáčknutí a vývojář má možnost definovat, co se při takové události odehraje.
- 2) *Nativní procedury* – tyto procedury lze vyvolat kdekoliv v projektu a měnit vlastnosti přístroje. Názorným příkladem je procedura *Hide()* u přístroje *Button*, která při zavolání skryje přístroj.
- 3) *Uživatelské procedury* – tyto procedury jsou vytvářeny uživatelem a k jejich aktivaci je musíme volat v aplikaci.

Každé *Uživatelské proceduře* může uživatel nadefinovat vstupní parametry a formát výstupních dat, ani jedno však není povinné. Uvnitř každé procedury je možné také definovat vlastní proměnné.

Všechny přístroje mají některé parametry, *Nativní procedury* a *Událostní procedury* společné. Vývojové prostředí navíc nabízí dědičnost vlastností, takže lze vytvořit šablonu přístroje a dále vytvářet jeho potomky. Strom těchto dědičností se zobrazuje v okně *Předlohy*.

2.2.2 Příklady přístrojů

Control Web má velice širokou škálu přístrojů. Jsou zde přístroje pro řízení, vizualizaci, archivaci a jiné práce s daty. Pro příklad uvedeme některé.

Meter – přístroj pro jednoduché zobrazování číselných dat. Má možnost tato data zobrazovat několika způsoby: digitálně, analogově, v grafu.

Control – pomocí tohoto přístroje lze nastavovat hodnotu číselné proměnné. Také tento přístroj má více módů: vertikální/horizontální posuvník, otočné kolečko, přímý zápis hodnoty.

Button – přístroj pracuje jako jednoduché tlačítko, kde lze reagovat na jeho zmáčknutí v proceduře *OnPress()*.

PID regulátor – zajímavý přístroj, který plně zastupuje funkci PID regulátoru. Nastavuje se proporcionální, integrační a derivační konstanta. Připojuje se žádaná, regulovaná a akční veličina. Podle schématu regulace se počítá z přírůstkového tvaru rovnice (rovnice 1) číslicového regulátoru z žádané a regulované hodnoty hodnota akčního zásahu.

$$u_{k+1} = u_k + r_0 \left[e_{k+1} - e_k + \frac{T_v}{T_i} e_{k+1} + \frac{T_d}{T_v} (e_{k+1} + 2e_k + e_{k-1}) \right] \quad (1)$$

Program – přístroj, který je nutný vložit mezi neviditelné přístroje. Samotný přístroj nic nedělá, používá se však k programování procedur, které pak chceme při běhu vyvolávat jinými přístroji.

File – přístroj, který je také nutný vložit mezi neviditelné přístroje, zajímavým je pro své procedury, které umí pracovat se soubory.

Core – je zvláštním typem přístroje. Nelze ho přidat, protože již je obsažen v každé aplikaci. Umí zasahovat do chodu programu. Zajímavou procedurou je například *StopApplication()*, která bezpodmínečně ukončí běh aplikace.

Sequencer – přístroj pro časování ostatních přístrojů. Tomuto přístroji je možné nastavit jeho vzorkovací periodu a poté už jen nastavovat přístrojům, aby se časovaly v závislosti na běhu tohoto časovače.

2.3 Simulace v systému Control Web

Jelikož ne vždy je možné odladit program, ve kterém pracujeme s nějakým systémem, v reálných podmínkách na fyzickém zařízení, je rozhodně vítanou možností si celý proces odsimulovat virtuálně. K tomuto účelu je v prostředí Control Web připraven modelový ovladač.

2.3.1 Modelový ovladač

Modelový ovladač je schopen virtuálně nahradit fyzikální systém a lze ho pro účel simulace řídicího pochodu použít. K práci s ovladačem nám prakticky stačí pouze

nadefinovat dva kanály – vstupní a výstupní. Vstupním kanálem budíme simulovaný systém a kanálem výstupním přichází hodnota reakce.

U tohoto modelového ovladače máme navíc možnost nastavit jeho parametry. V dialogovém okně, které se nám automaticky ukáže při startu aplikace, lze nastavit řád systému a parametry jeho matematického modelu. Je však možné nastavit nejvýše soustavu 4. řádu. Navíc je zde možné také nastavit, zda bude signál zašuměný a popřípadě jeho rozptýl. Nevýhodou je, že nelze nastavit dopravní zpoždění systému.

2.3.2 Nástroj pro ladění aplikace

Užitečným nástrojem pro vývojáře je rozhodně nástroj *Pro ladění aplikací*. Při spuštění aplikace se zapíná *Okno sledování a zpráv*. Tento nástroj se dělí na 3 části.

V horní liště se nacházejí tlačítka pro řízení běhu aplikace. Je možné aplikaci zastavit, znovu spustit, krokovat (po krocích dané jádrem systému), periodicky zastavovat běh a dále je možné zastavit či povolit periodické obnovování dat.

V horní části okna je možné pomocí rozklikávacího menu zobrazit některé údaje běžící aplikace. Je možné zobrazit stav jakéhokoliv datového elementu kromě těch, které jsou definovány přímo v procedurách přístrojů. Dále je možné sledovat stav přístrojů, například kolikrát již byly aktivovány a další jiné údaje především ohledně časování přístrojů. Lze si také zobrazit stav ovladačů, jaký je aktuální vstup, výstup.

V dolní části okna se nacházejí 4 záložky zpráv:

- *Chybové zprávy* – zobrazují informace o chybách při překladu či běhu aplikace.
- *Ladicí zprávy* – zde jsou vypisovány zprávy aplikací nebo jejichmi komponentami (ovladače aj.).
- *Informativní zprávy* – zde se často ukazují zprávy doplňující chybová hlášení, navíc jsou zde informace o proběhlém překladu, spuštění či zastavení aplikace.
- *Systémové zprávy* – zde jsou doplňkové informace, které nesouvisí s aplikacemi.

3. Identifikace dynamických systémů

V této kapitole se budeme zabývat identifikací reálných systémů. Zmíníme se o metodě 1. řádu s dopravním zpožděním a metodě nejmenších čtverců. Při psaní této kapitoly vycházíme z učebního textu *Identifikace systémů* od Vrožiny, Jančíkové a Davida a z publikace *Modelování a identifikace* od Noskieviče.

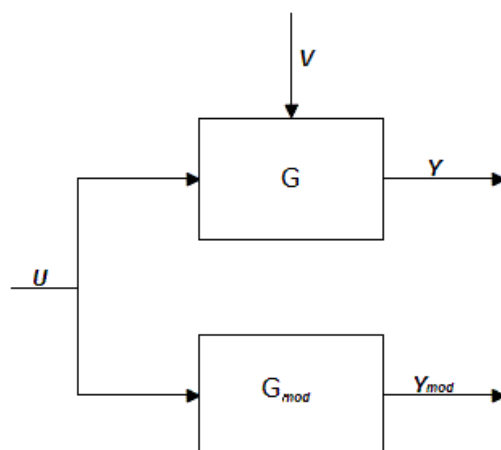
3.1 Identifikace

Pojmem identifikace v oblasti řízení rozumíme ztotožnění reálného systému s jeho modelem. Identifikací získáváme strukturu a parametry modelu, který popisuje právě identifikovaný systém. Tato informace je nezbytná ve chvíli, kdy chceme daný systém regulovat. Navíc pomocí určeného modelu lze simulovat chování systému, tedy například určovat reakce na vstupní signály.

Je velice důležité nezapomínat, že při jakékoliv identifikaci se vždy dopouštíme chyby oproti reálnému systému. Získáváme pouze aproximaci systému, která plně neodpovídá reálnému systému. Výsledný model je tedy vždy zjednodušením.

Systémem rozumíme reálný objekt, který interaguje se svým okolím. Má svůj měřitelný vstup i výstup. Vstupem systému je vektor $U = (u_1, u_2, \dots, u_n)$ – soubor všech měřitelných vstupů. Výstupem je vektor $Y = (y_1, y_2, \dots, y_m)$, který představuje soubor všech měřitelných výstupů. Do každého reálného systému navíc vstupuje vektor poruchy $V = (v_1, v_2, \dots, v_k)$, který představuje všechny poruchové vstupy.

Operátor G na schématu představuje reálný systém, kterému se budeme snažit přiblížit. Na obrázku 4 jsou zobrazeny systém i jeho model. Model může být vyjádřen různými formami, například matematickou funkcí, tabulkou, grafem. My hledáme matematickou funkci.



Obrázek 4: Schéma modelu

Při procesu identifikace za použití stejných vstupů U hledáme co nejpodobnější odezvu modelu $Y_{mod} = (y_{mod1}, y_{mod2}, \dots, y_{modm})$ odezvě systému Y , abychom našli model soustavy G_m . Pokud by rozdíl $Y - Y_{mod}$ byl nulový, můžeme říct, že výstup modelu se chová stejně jako výstup systému.

3.1.1 Matematicko-fyzikální metody identifikace

Matematicko-fyzikální metoda je založena na znalosti vnitřního uspořádání soustavy. Pomocí matematicko-fyzikální analýzy se sestaví matematický model, který vychází z fyzikálních vlastností soustavy. Problémem této metody je fakt, že složitost matematického řešení roste se složitostí soustavy a potřebnou přesností modelu. Protože ale nikdy není možné zajistit plně věrný model, musí docházet k jistým zjednodušením modelu vůči soustavě. Výhodou této metody je jasně daný vztah mezi parametry modelu a vnitřními parametry soustavy. Další výhodou je bezesporu možnost odsimulovat chování soustavy na základě znalosti popisu dříve, než je reálný systém vůbec spuštěn. To může být žádoucí ve chvíli, kdy není možné spouštět systém bez předchozí analýzy jeho chování. Modelu, který vznikne touto metodou identifikace, se říká *white box*.

3.1.2 Experimentální metody identifikace

Oproti matematicko-fyzikální analýze vychází experimentální metody identifikace z již naměřených dat reálného systému. Výsledkem této analýzy je model vyjadřující vnější popis chování soustavy (takzvaný *black box*). Matematický model vyjadřuje pouze vztah mezi vstupem a výstupem, neřeší vnitřní složitost soustavy. Při procesu identifikace je velice důležité zvolit správný matematický model (strukturu), aby výsledek byl co nejpřesnější. Experimentálních metod v historii vznikla celá řada, lze je dělit podle různých kritérií.

Protože jsou vstupní signály děleny na deterministické a stochastické, kde deterministické jsou jasně popsateľné funkce času a stochastické jsou náhodné, nepopsateľné funkcí času, tak jsou pak také děleny metody identifikace na deterministické a stochastické.

Reálné systémy jsou většinou ve spojitě oblasti, proto první experimentální metody vznikaly v časové oblasti dat, kde pro popis systému slouží diferenciální rovnice. Z důvodu rozvoje počítačové techniky začaly vznikat diskretizované modely. Přechod na počítače umožnil rychlejší výpočet s dostatečnou přesností. V diskretní oblasti se pro popis systému používá diferenční rovnice.

Podle způsobu zpracování naměřených dat můžeme dělit ještě na online a offline metody. Online metodami rozumíme postupné zpracování dat, kde se již z několika hodnot zjišťují parametry modelu. Z dalších hodnot se model pouze zpřesňuje. Výhodou těchto metod je, že není potřeba mít archivované všechny naměřené hodnoty. Offline metodami rozumíme metody, kde je výpočet matematického modelu proveden až po ukončení měření. Pro tyto metody jsou potřeba všechna data měření.

Pomocí experimentálních metod můžeme identifikovat do takzvaných parametrických, či neparametrických modelů. Parametrickými modely rozumíme model s danou strukturou. Tuto strukturu je potřeba znát dopředu (například řád diferenciální rovnice) nebo ji vhodně odhadnout. U neparametrických modelů neznáme jejich strukturu, jsou určeny grafem nebo tabulkou. Neparametrický model nelze použít přímo pro řízení, proto je potřeba vhodně parametrizovat.

3.2 Struktura lineárních modelů

Většina existujících systémů je nelineárních, ty jsou často zjednodušovány a linearizovány do soustav lineárních. Právě těmi se v práci budeme zabývat.

Volba vhodného lineárního modelu je velice důležitou součástí pro proces identifikace a výslednou přesnost.

3.2.1 Lineární model ve spojitém čase

Obecný lineární systém je ve spojitém čase popsán rovnicí:

$$a_n y^{(n)} + \dots + a_1 y^{(1)} + a_0 y = b_m u^{(m)} + \dots + b_1 u^{(1)} + b_0 u \quad (2)$$

Kde $y^{(i)}$ jsou derivace výstupu stupně i , $u^{(j)}$ jsou derivace vstupu až stupně j a a_i a b_j jsou konstantními koeficienty. Podmínkou fyzikální realizovatelnosti je $n \geq m$.

Pro určení přenosu takovéto lineární diferenciální rovnice je potřeba zavést Laplaceův obraz vstupu a výstupu. Přenos v Laplaceově obraze je tedy podíl Laplaceova obrazu výstupu k Laplaceova obrazu vstupu.

$$G(s) = \frac{Y(s)}{U(s)} = \frac{b_m s^m + \dots + b_1 s + b_0}{a_n s^n + \dots + a_1 s + a_0} \quad (3)$$

3.2.2 Tvary lineárních modelů v diskrétním čase

Obecný tvar lineárního parametrického diskrétního modelu s pouze jedním vstupem i výstupem (takzvaný SISO model) je znázorněn rovnicí 4 podle Noskiewiče⁶.

$$y(k) + a_1 y(k-1) + \dots + a_n y(k-n) = b_0 u(k) + b_1 u(k-1) + \dots + b_n u(k-n) \quad (4)$$

Koeficienty a_1, a_2, \dots, a_n a b_0, b_1, \dots, b_n jsou parametry modelu. Úkolem identifikace je jejich zjištění. Výstupem je $y(k)$ a $u(k)$ je vstupem modelu v diskrétním čase k .

⁶ NOSKIEWIČ, P.: Modelování a identifikace systémů. MONTANEX a.s., 1999, ISBN 80-7225-030-02

Koeficient a_0 bývá normalizován na hodnotu 1, což umožňuje výhodně určit hodnotu výstupu $y(k)$, protože při numerickém řešení není třeba neustále dělení koeficientem a_0 .

Výstup modelu v diskrétním čase k se poté určí takto:

$$y_m(k) = - \sum_{i=1}^n a_i y_m(k-i) + \sum_{i=1}^m b_i u(k-i) \quad (5)$$

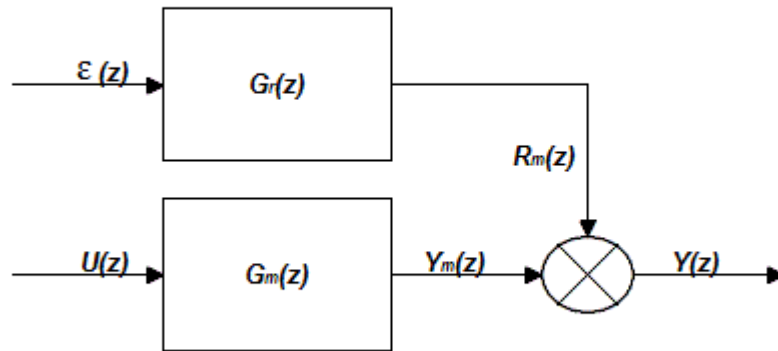
Po aplikaci Z-transformace vypadá rovnice takto:

$$Y_m(z) = -Y_m(z)(a_1 z^{-1} + \dots a_n z^{-n}) + U(z)(b_0 + b_1 z^{-1} + \dots b_m z^{-m}) \quad (6)$$

A Z-přenos $G(z)$ získáme tedy jako:

$$G(z) = \frac{Y_m(z)}{U(z)} = \frac{b_0 + b_1 z^{-1} + \dots b_m z^{-m}}{1 + a_1 z^{-1} + \dots a_n z^{-n}} \quad (7)$$

Pokud chceme vzít v úvahu měřitelnou chybu výstupu, musíme podle publikace *Modelování a identifikace* upravit model a přidat poruchový signál, který působí na výstup systému. To je znázorněno na obrázku 5.



Obrázek 5: Upravený model s náhodným poruchovým signálem

$\varepsilon(z)$ zde představuje bílý šum.

$R_m(z)$ je výstupní poruchový signál, který je možné vyjádřit vzorcem:

$$R_m(z) = G_r(z) \varepsilon(z) \quad (8)$$

Pokud je struktura modelu zvolena vhodně, výstupní signál modelu se blíží výstupnímu signálu systému.

$$y(k) = y_m(k) + r_m(k) \quad (9)$$

A dále po dosazení za $r_m(k)$ dostaneme přenos v Z-transformaci

$$Y(z) = Y_m(z) + G_r(z) \varepsilon(z) \quad (10)$$

Když nyní budeme měnit matematickou strukturu $G_r(z)$, tak dostaneme různé struktury lineárních modelů.

Zavedeme polynomy:

$$A(z^{-1}) = 1 + a_1 z^{-1} + \dots a_n z^{-n} \quad (11)$$

$$B(z^{-1}) = b_0 + b_1 z^{-1} + \dots b_m z^{-m} \quad (12)$$

Pokud $G_r(z) = \frac{1}{A(z^{-1})}$,

tak model nazýváme modelem ARX (AutoRegresive model with eXternal input). Jeho popis tedy je:

$$Y(z) = \frac{B(z^{-1})}{A(z^{-1})} U(z) + \frac{1}{A(z^{-1})} \varepsilon(z) \quad (13)$$

Dále pokud zavedeme polynomy:

$$C(z^{-1}) = 1 + c_1 z^{-1} + \dots c_n z^{-n} \quad (14)$$

$$D(z^{-1}) = 1 + d_1 z^{-1} + \dots d_m z^{-m} \quad (15)$$

Tak lze vyjádřit, že $G_r(z) = \frac{C(z^{-1})}{A(z^{-1})}$, v tomto případě se modelu říká *ARMAX* (*Average model with eXternal input*).

Pokud $G_r(z) = \frac{C(z^{-1})}{D(z^{-1})}$, tak se modelu říká *BJ* (*Box-Jenkins*).

A pokud $G_r(z) = 1$, tak je model nazýváme *OE* (*Output error*).

3.3 Vybrané metody identifikace

Jak již bylo zmíněno, v historii vzniklo mnoho identifikačních metod založených na různých principech. Po dohodě s vedoucím práce jsme vybrali dvě metody k určení modelu systému výparníku. Obě dvě metody jsou určeny k identifikování modelu prvního řádu. První řád jsme zvolili, protože naměřené charakteristiky byly prvním řádem velmi dobře aproximovatelné (jak je videt na grafu 1) a sám výrobce v dokumentaci uvádí aproximaci prvním řádem jako typickou metodu pro identifikaci soustavy výparníku.

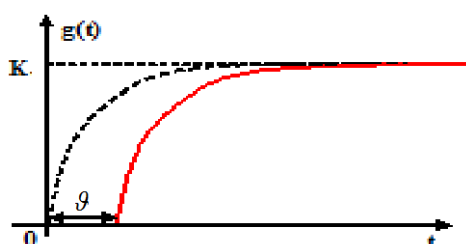
3.3.1 Aproximace 1. řádem s dopravním zpožděním

Nejjednodušší metodou obecně pro identifikaci stabilní, nekmitavé charakteristiky je metoda aproximace 1. řádem. Tato metoda je sice primitivní, avšak právě pro soustavy 1. řádu velice šikovná.

Obecný tvar přenosu soustavy 1. řádu s dopravním zpožděním je:

$$G(s) = \frac{k}{Ts+1} e^{-\vartheta s} \quad (16)$$

Kde ϑ představuje dopravní zpoždění znázorněné na obrázku 6.

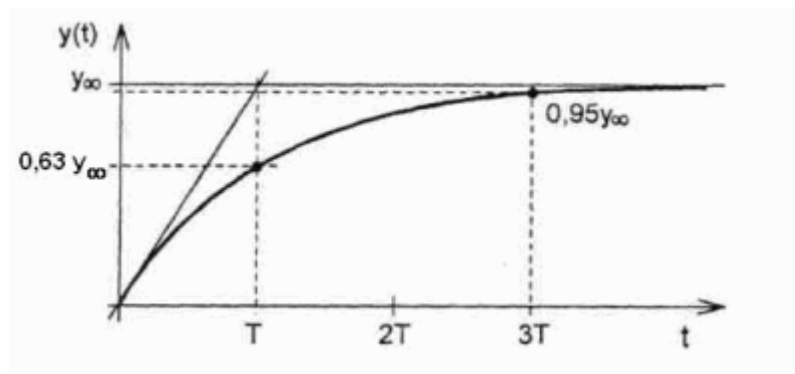


Obrázek 6: Dopravní zpoždění

K aproximaci musíme mít již data přechodové charakteristiky s ustálenou hodnotou výstupu systému. Zesílení systému k se vypočítá jako poměr vstupního signálu na ustálený výstupní signál.

$$k = \frac{y(\infty)}{u(\infty)} \quad (17)$$

Časová konstanta T se určí jako čas ve chvíli, kdy přechodová funkce dosáhla 63% ustálené hodnoty nebo třetina času 95% ustálené hodnoty.



Obrázek 7: Aproximace prvním řádem, určení časové konstanty; obrázek převzat ze skript Identifikace systémů ⁷

3.3.2 Metoda nejmenších čtverců

Metoda nejmenších čtverců je matematickou metodou aproximace, která umožňuje proložit co nejpřesněji vstupní data (body) křivkou zvoleného typu. Tímto typem křivky může být například přímka, logaritmická křivka či polynom zvoleného stupně. Pro aproximaci je použito kvadratické kritérium pro zjištění minimální odchylky od prokládaných bodů. Identifikace touto metodou vede do diskrétního času.

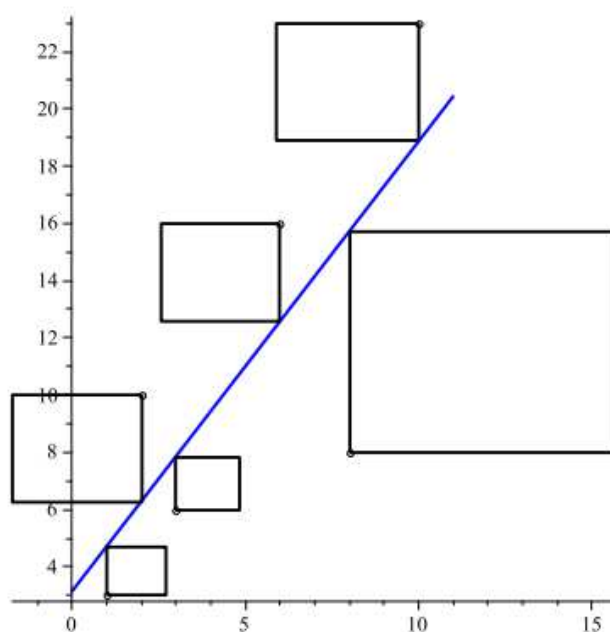
Pro názornost nejprve uvedeme použití metody nejmenších čtverců v případě aproximace přímky.

Aproximace přímkou

Předpokládejme, že máme dány body $[x_0, y_0], [x_1, y_1], \dots, [x_n, y_n]$ a hledáme funkci typu $y = a + bx$, která aproximuje tyto body, tím vzniká přeuročená soustava rovnic.

⁷ VROŽINA, Milan, Zora JANČÍKOVÁ a Jiří DAVID. TECHNICKÁ UNIVERZITA OSTRAVA. *Identifikace systémů*. Ostrava, 2012. ISBN 978-80-248-2594-6.

Jinak řečeno hledáme přímku, která bude mít nejmenší součet čtverců odchylek zadané souřadnice y_i a souřadnice $a + bx_i$.



Obrázek 8: Názorný obrázek aproximace; obrázek převzat ze skript Matematika 3 pro FEKT VUT⁸

$$E(a, b) = \sum_{i=0}^n (y_i - (a + b x_i))^2 \quad (18)$$

Kde $E(a, b)$ je odchylka závislá na parametrech a a b . Její minimum lze nalézt pomocí první derivace podle parametrů a a b položené rovno nule. Tedy: $\frac{\partial E}{\partial a} = 0$ a $\frac{\partial E}{\partial b} = 0$.

$$\sum_{i=0}^n (2(y_i - a - bx_i)(-1)) = 0 \quad (19)$$

$$\sum_{i=0}^n (2(y_i - a - bx_i)(-x_i)) = 0 \quad (20)$$

⁸ Fajmon, B., Růžicková, I.: Matematika 3. Skriptum FEKT VUT v elektronické formě, Brno 2013.

Z rovnice 19 a 20 lze dosáhnout tvarů:

$$a n + b \sum_{i=0}^n x_i = \sum_{i=0}^n y_i \quad (21)$$

$$a \sum_{i=0}^n x_i + b \sum_{i=0}^n x_i^2 = \sum_{i=0}^n (x_i y_i) \quad (22)$$

Tímto nám vznikly prakticky dvě rovnice o dvou neznámých. Z rovnice lze získat parametry a a b .

Často při výpočtech na počítači je mnohem výhodnější maticový zápis. Vezměme přeurčenou soustavu rovnic:

$$\begin{aligned} y_0^* &= a + bx_0 + e_0 \\ y_1^* &= a + bx_1 + e_1 \\ &\vdots \\ y_n^* &= a + bx_n + e_n \end{aligned} \quad (23)$$

Maticově lze zapsat:

$$\begin{pmatrix} y_0^* \\ y_1^* \\ \vdots \\ y_n^* \end{pmatrix} = \begin{pmatrix} 1 & x_0 \\ 1 & x_1 \\ \vdots & \vdots \\ 1 & x_n \end{pmatrix} * \begin{pmatrix} a \\ b \end{pmatrix} + \begin{pmatrix} e_0 \\ e_1 \\ \vdots \\ e_n \end{pmatrix} \quad (24)$$

V označení vektorů a matice je:

$$\mathbf{y}^* = \mathbf{Z} \cdot \mathbf{p} + \mathbf{e} \quad (25)$$

Kde \mathbf{y}^* je výstupní vektor se zadanými y souřadnicemi, \mathbf{p} je vektor parametrů a \mathbf{Z} je matice s hodnotami x_i a vektor \mathbf{e} je vektor s chybami rovnic e_0, e_1, \dots, e_n .

Je možné dokázat, že maticový zápis rovnice pro výpočet parametrů a a b je:

$$\mathbf{p} = (\mathbf{Z}^T \mathbf{Z})^{-1} \mathbf{Z}^T \mathbf{y} \quad (26)$$

Odhlad parametrů modelu ARX metodou nejmenších čtverců

V této části práce se budeme zabývat modelem ARX, kde budeme zjišťovat parametry potřebné k identifikaci. Model ARX má diferenční rovnici:

$$y(k) = -a_1 y(k-1) - \dots - a_n y(k-n) + b_0 u(k-1) + \dots + b_m u(k-m) + e(k) \quad (27)$$

V minulé kapitole jsme určili maticovou rovnici (26) pro výpočet parametrů pomocí metody nejmenších čtverců. Tato rovnice je obecně platná i pro aproximaci polynomem, musí se ovšem upravit matice \mathbf{Z} , správně definovat vektor \mathbf{y} a upravit vektor parametrů \mathbf{p} .

Matice \mathbf{Z} bude pro aproximaci 1. řádu pomocí metody nejmenších čtverců:

$$\mathbf{Z} = \begin{pmatrix} y_0 & u_i \\ y_1 & u_{i+1} \\ \vdots & \vdots \\ y_n & u_{n-i} \end{pmatrix} \text{ a vektor } \mathbf{y} \text{ je definován jako: } \mathbf{y} = \begin{pmatrix} y_1 \\ y_2 \\ \vdots \\ y_{n-1} \end{pmatrix} \text{ a vektor } \mathbf{p} = \begin{pmatrix} -a \\ b \end{pmatrix},$$

kde $i \geq 0$. Index i je zde zaveden kvůli zjištění dopravního zpoždění.

Navíc je důležité si zavést vektor \mathbf{e} , kterým určíme chybu modelu: $\mathbf{e} = \begin{pmatrix} e_1 \\ e_2 \\ \vdots \\ e_n \end{pmatrix}$.

Vektor \mathbf{e} se vypočítá jako: $\mathbf{e} = \mathbf{y} - \mathbf{Zp}$ a chyba modelu ε je rovna $\varepsilon = \mathbf{e}^t \mathbf{e}$.

Pro zjištění dopravního zpoždění metoda vyhodnotí parametry vícekrát se vstupním signálem u posunutým o i , podle toho v jakém průběhu bude chyba modelu ε nejmenší, se vyhodnotí dopravní zpoždění.

Model pro identifikaci prvním řádem má tvar:

$$G_m(z) = \frac{b}{z+a} \quad (28)$$

Dopravní zpoždění se poté zavede vynásobením přenosu z^{-i} .

4. Návrh konstant PID regulátoru

V regulačních soustavách se nejčastěji používají PID regulátory. Takový regulátor obsahuje i karta pro napařování tenkých vrstev. Možnosti, jak navrhnout PID regulátor, jsou různé. Historicky první metodou je metoda Zieglera a Nicholse z roku 1942. Dnes již jsou popsány další desítky metod, které pracují na základě předpisů a empirických tabulek.

Po dohodě s vedoucím práce jsme se rozhodli použít metodu, která určuje hodnoty pro nastavení PID regulátoru na principu výpočtu z celkového regulačního pochodu. Této metodě se říká přímá syntéza regulátoru, kterou v kapitole popíšeme. Dále se zmíníme o metodě návrhu konstant PID regulátoru, kterou nabízí výrobce PCI karty *SQM242*, jež je popsána v dokumentaci karty. Pracujeme s přenosem soustavy prvního řádu, to doporučuje výrobce karty a především reálná naměřená data prvnímu řádu nejvíce odpovídají.

Při identifikaci metodou nejmenších čtverců určujeme parametry diskrétního modelu. Ovšem PID regulátor karty je přizpůsoben na konstanty určené z modelu v Laplaceově transformaci. Musíme tedy nejprve převést z modelu v Z transformaci na model v Laplaceově transformaci.

4.1 Převod ze Z obrazu do Laplaceova obrazu

Vezmeme přenos modelu $G(z) = \frac{0,2970}{z-0,9802}$ s periodou vzorkování 0,1 s, pro zjištění jeho zesílení stačí dosadit za všechna z v modelu hodnoty jedna. Konstanta zesílení tedy je $k = 15$.

Důvodem je, věta o konečné hodnotě, která říká, že: $\lim_{t \rightarrow \infty} y(t) = \lim_{s \rightarrow 0} s Y(s)$, tedy rovnice $z = e^{sT}$, kde s je komplexní proměnná a T je vzorkovací perioda, se rovná jedné pouze ve chvíli, kdy je s rovno nule, tedy v momentě, kdy je signál ustálen.

Pro zjištění časové konstanty z modelu $G(z)$ musíme vzít předpoklad ze slovníku z Laplaceova obrazu do Z obrazu:

$$\frac{1}{s+a} = \frac{z}{z-e^{-aT}} \quad (29)$$

Nyní můžeme říct, že $0,9802 = e^{-aT}$. Po úpravě a dosazení periody vzorkování vyjde vztah:

$$a = -\left(\frac{\ln(0,9802)}{0,1}\right) \quad (30)$$

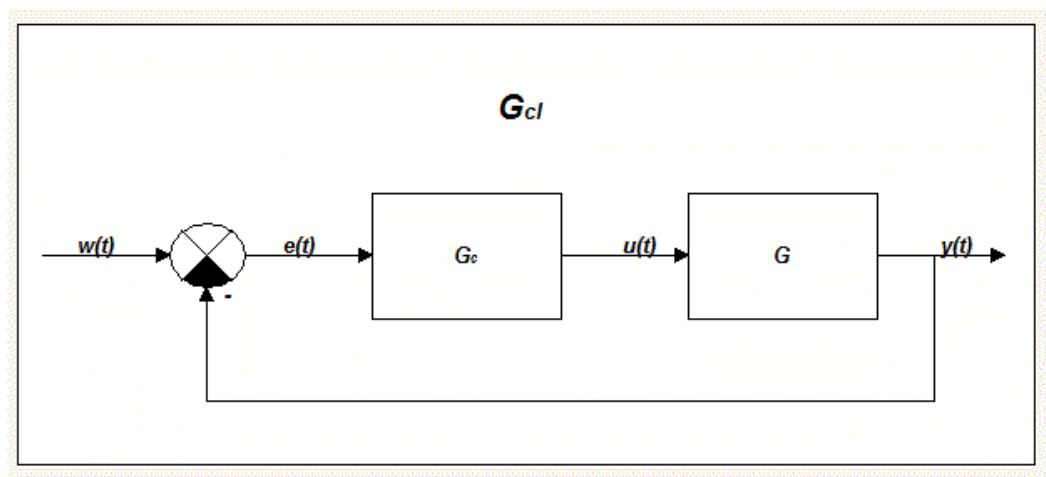
Časová konstanta $\tau = a^{-1} = 5$.

Výsledný model má tedy tvar:

$$G(s) = \frac{15}{5s+1} \quad (31)$$

4.2 Přímá syntéza regulátoru

Přímá syntéza regulátoru je založena na návrhu žádaného přenosu celkového regulačního pochodu. Pomocí matematických úprav a dosazení identifikovaného modelu dostáváme prakticky paralelní tvar rovnice PID regulátoru, ze kterého lze odečíst jeho konstanty. Tuto metodu popisuje Klán v časopise *Automa*⁹.



Obrázek 9: Schéma regulačního pochodu

⁹ KLÁN, Petr. O přímé syntéze PID regulátorů. *Automa*. 2006, č. 11.

Přenos regulačního pochodu lze pomocí blokové algebry určit vzorcem:

$$G_{cl} = \frac{GG_c}{1+GG_c} = \frac{1}{G} \left(\frac{\frac{Y}{W}}{1-\frac{Y}{W}} \right), \quad (32)$$

kde G_{cl} je přenos celého regulačního pochodu, G je přenos regulované soustavy, G_c je přenos PID regulátoru, Y je Laplaceova transformace regulované veličiny, W je Laplaceova transformace žádané hodnoty.

Z rovnice (32) lze vyjádřit G_c , tedy přenos PID regulátoru. Protože ovšem neznáme skutečné veličiny G a Y/W , je lepší používat veličiny G_m , Y_m/W_m , tedy veličiny modelu procesu. Vztah pro výpočet G_c vypadá takto:

$$G_c = \frac{1}{G_m} \left(\frac{\frac{Y_m}{W_m}}{1-\frac{Y_m}{W_m}} \right) \quad (33)$$

Když zvolíme, že $\frac{Y_m}{W_m} = \frac{1}{\tau_p s + 1}$, určíme, že se bude regulační proces chovat jako soustava prvního řádu a přidáme dopravní zpoždění, tak přenos regulátoru vyjádří vzorec:

$$G_c = \frac{1}{G_m} \left(\frac{e^{-\theta s}}{(\tau_p s + \theta)s} \right) \quad (34)$$

Nyní, když zvolíme jako systém G_m systém prvního řádu s dopravním zpožděním a dosadíme do rovnice (34), tak po úpravách vyjde rovnice PI regulátoru.

$$G_m = \frac{k e^{-\theta s}}{\tau s + 1} \quad (35)$$

$$G_c = \frac{1}{k} \left(\frac{\tau}{\tau_p + \theta} \right) \left(1 + \frac{1}{\tau s} \right) \quad (36)$$

Zavedeme konstanty zesílení regulátoru K_c a integrační konstantu T_i . Ty jsou určeny jako:

$$K_c = \frac{1}{k} \left(\frac{\tau}{\tau_p + \theta} \right) \quad (37)$$

$$T_i = \tau \quad (38)$$

Tímto máme určené konstanty regulátoru. Je nyní na zvážení, zda by byla nebo nebyla potřeba derivační složka PID regulátoru.

4.3 Výrobce zadané vztahy pro určení konstant PID regulátoru

Výrobce karty doporučuje, pro určení konstant PID regulátoru z identifikovaného modelu soustavy, tyto hodnoty:

$$K_c = \left(\frac{1,36}{k}\right) \left(\frac{\vartheta}{\tau}\right)^{-0,947} \quad (39)$$

$$T_i = (1,19\tau) \left(\frac{\vartheta}{\tau}\right)^{0,738} \quad (40)$$

$$T_d = (0,381\tau) \left(\frac{\vartheta}{\tau}\right)^{0,995} \quad (41)$$

Výrobce vycházel při návrhu konstant PID regulátoru integrálního kritéria *ITAE* (*Integral of Time Absolute Error*)

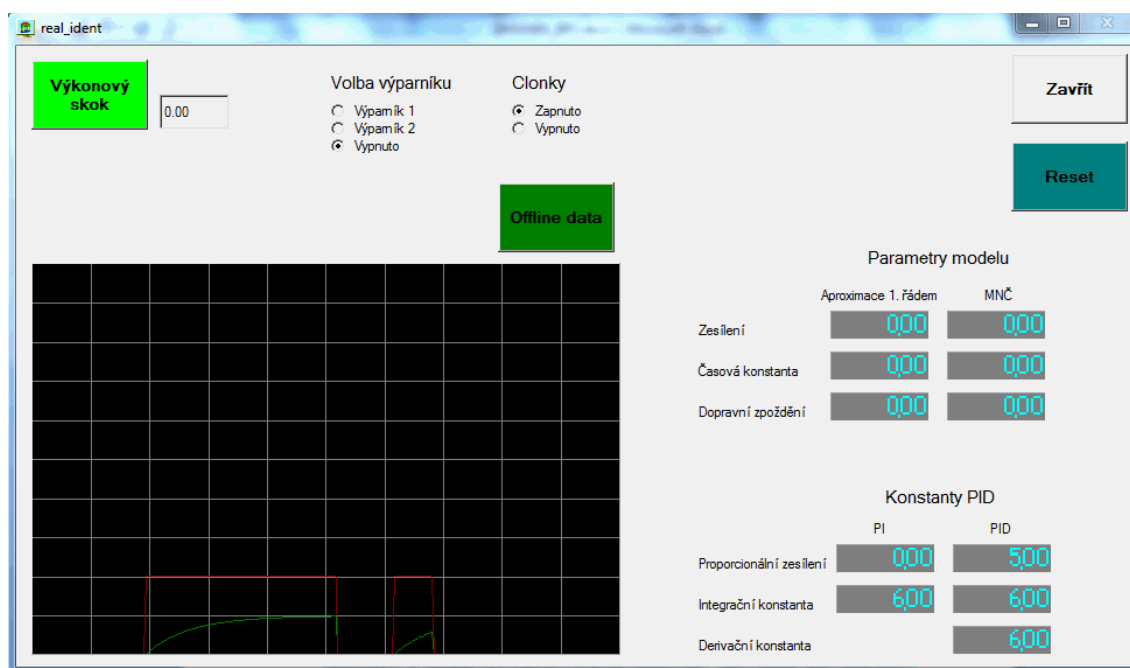
$$ITAE = \int t|\varepsilon|dt \quad (42)$$

5. Modul pro identifikaci

Ve vývojovém prostředí Control Web jsme vytvořili modul pro identifikaci soustavy. Bohužel tento modul nemohl být vyzkoušen na fyzické napařovací aparatuře, protože v době tvorby této práce nebyla funkční karta pro napařování tenkých vrstev. Celý proces identifikace je tedy proveden simulačně pomocí modelového ovladače. V modulu je navíc možné nahrávat již naměřená data ze souboru a ty poté identifikovat. Tento modul není samostatnou aplikací, slouží pro připojení do aplikace vytvořené v prostředí Control Web.

5.1 Popis modulu

Modul je určen k identifikaci odporového výparníku napařovací aparatury. Tato identifikace je důležitá pro regulaci rychlosti napařování. Jako vstupní data identifikace je použita hodnota řídicího napětí vstupujícího do regulátoru výkonu a jako výstupní data je použita hodnota rychlosti napařování.



Obrázek 10: Uživatelské rozhraní modulu identifikace

Na obrázku 10 je zobrazeno uživatelské rozhraní modulu identifikace. Toto rozhraní je upraveno tak, aby splňovalo základní požadavky pro potřebu obsluhy. Tlačítko *Výkonový skok* slouží k přivedení výkonového skoku na odporový výparník. V modulu je tato možnost realizována pomocí procedury, která zašle příkaz kartě s hodnotou výkonu v rozmezí nula až jedna, aby nastavila svůj napěťový výstup na odpovídající hodnotu. Hodnota výkonového skoku v rozmezí od nuly do jedné se nastaví v okně vedle tlačítka a tlačítkem se potvrdí vyslání. Před stisknutím tlačítka již musí být nastaveno, který odporový výparník budeme identifikovat, toto nastavení se provede pomocí výběru *Volba výparníku*, tato volba mění nastavení stykače, který volí, do kterého výparníku bude přiveden proud. Stykač má navíc polohu, kde není nastaven ani jeden z výparníků. Po stisknutí tlačítka *Výkonový skok* se pod ním objeví tlačítko *Hodnota ustálena*, které je určeno pro to, aby obsluha určila, zda je již hodnota rychlosti proudu ustálena. Po stisknutí tohoto tlačítka se vypočítají procedury pro identifikaci a následné určení konstant PID regulátoru z naměřené přechodové charakteristiky. Tyto hodnoty se zobrazí v kolonkách *Parametry modelu* a *Konstanty PID*. Po stisknutí tlačítka *Hodnota ustálena* tlačítko opět zmizí a je možné zadat další výkonový skok. Po dalším následném vyhodnocení parametru modelu a konstant PID regulátoru jsou hodnoty zprůměrovány.

Modul nabízí možnost nahrávat již naměřené přechodové charakteristiky. To se děje pomocí tlačítka *Offline data*, které po stisknutí vyvolá procedury potřebné k nahrání dat. Tato data jsou identifikována a z nich se vypočítají parametry PID regulátoru. Tyto hodnoty jsou následně vypsány opět do kolonek *Parametry modelu* a *Konstanty PID*.

Na ploše modulu se nacházejí i tlačítka *Reset* a *Zavřít*. Tlačítko *Reset* slouží k vynulování kolonek *Parametry modelu* a *Konstanty PID* a k vymazání vnitřních proměnných. Tlačítko *Zavřít* zavírá celou aplikaci.

5.2 Proces identifikace

Proces identifikace zahrnuje několik potřebných akcí, které v této podkapitole popíšeme. Nejprve je potřeba daný materiál zahřát na teplotu, kdy se začne odpařovat. Při tomto zahřívání je potřeba, aby byly nad výparníky zataženy clonky, protože

z některých materiálů při zahrávání odlétávají částičky a ty by mohly znehodnotit substrát. Po zahřátí na správnou teplotu mohou být teprve provedeny výkonové skoky, které provádí modul pro identifikaci. Při tomto procesu jsou již clonky odtaženy, aby mohla být měřena rychlost napařování. Výkonové skoky mohou být provedeny jak kladné, tak záporné. Data jsou čtena z kanálů každou sekundu, to znamená, že naše perioda vzorkování je jednu sekundu.

5.3 Metody identifikace

Programování procedur v prostředí Control Web je zajímavou možností rozšíření funkčnosti aplikace. V této podkapitole jsou zmíněny některé procedury potřebné při identifikaci parametrů odporového výparníku použitými metodami.

Aproximace prvního řádu

Pro tuto metodu jsme vytvořili proceduru *Aproximace_radu_1(y,u: array of real)*. Je to jednoduchá procedura, do které vstupují dvě pole. Tato dvě pole jsou nejprve odečteny do nuly pomocí procedury *Odecti_do_nuly(y,u:array of real)*. To je zajištěno pomocí průměru tří změřených hodnot pole vstupů *u* a pole výstupů *y* před výkonovým skokem. Nejprve je pole výstupní filtrováno mediánovým filtrem, který vyvoláváme procedurou *Med_filtr(a: array off real)*. Poté je počítáno zesílení *k* jako podíl výstupní ustálené hodnoty *k* vstupní ustálené hodnotě. Časová konstanta je určena pomocí hodnoty 63,2 % ustálené hodnoty výstupu. K této hodnotě se najde nejbližší odpovídající časová hodnota. Jelikož se většinou tato hodnota nachází mezi časovými údaji, je potřeba ji zjistit lineární interpolací. Tím jsme získali časovou konstantu. Dopravní zpoždění je poté vypočítáno pouze jako rozdíl času, kdy nastal výkonový skok a začne reagovat soustava.

Metoda nejmenších čtverců

Pro tuto metodu je vytvořena procedura *Metoda_nejmensich_ctvercu(y,u: array of real)*. Na obrázku 11 je ukázán zdrojový kód procedury.

```

procedure Metoda_nejmensich_ctvercu( y : array of real; u : array of real );
var
  lATmp1 : array[0..1000] of real;
  lATmp2 : array[0..1000] of real;
  lTmp3 : real;
  lTmp4 : real;
  lTmp5 : real;
  lTmp6 : real;
  i : integer;
begin
  (*kvadrat matic*)
  for i = 0 to index_pole-2 by 1 do
    lATmp1[0] = lATmp1[0] + y[i]*y[i];
    lATmp2[1] = lATmp2[1] + u[i]*u[i];
    lATmp1[1] = lATmp1[1] + u[i]*y[i];
  end;
  lATmp2[0] = lATmp1[1];
  (*inverzni matice*)
  lTmp3 = lATmp2[1]*lATmp1[0] - lATmp1[1]*lATmp2[0];
  lTmp4 = lATmp1[0];
  lATmp1[0] = lATmp2[1]/lTmp3;
  lATmp2[1] = lTmp4/lTmp3;
  lATmp1[1] = -lATmp1[1]/lTmp3;
  lATmp2[0] = -lATmp2[0]/lTmp3;
  (*nasobeni se ZT*)
  lTmp3 = lATmp1[0];
  lTmp4 = lATmp1[1];
  lTmp5 = lATmp2[0];
  lTmp6 = lATmp2[1];
  for i = 0 to index_pole-2 by 1 do
    lATmp1[i] = lTmp3 * y[i] + lTmp4 * u[i];
    lATmp2[i] = lTmp5 * y[i] + lTmp6 * u[i];
  end;
  (*nasobeni s y*)
  for i = 0 to index_pole-2 by 1 do
    par_MNC[0] = par_MNC[0] + lATmp1[i]*(y[i+1]);
    par_MNC[1] = par_MNC[1] + lATmp2[i]*(y[i+1]);
  end;
  par_MNC[0] = -par_MNC[0];
end_procedure;

```

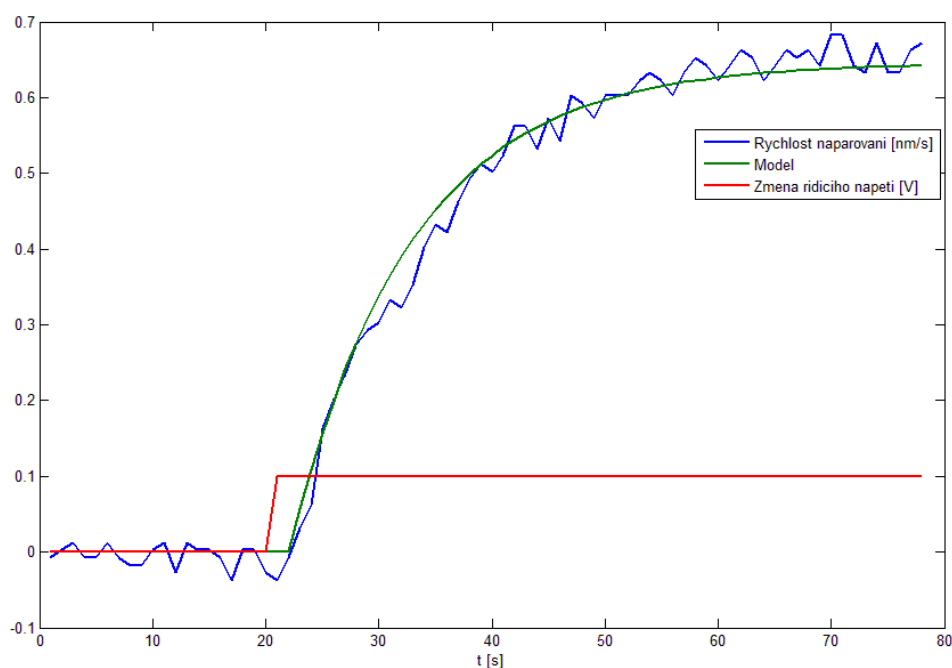
Obrázek 11: Metoda nejmenších čtverců implementovaná v proceduře

Pro tvorbu procedury byl použit vzorec $\mathbf{y} = (\mathbf{Z}^T \mathbf{Z})^{-1} \mathbf{Z}^T \mathbf{y}$. Pro maticové výpočty byly použity zjednodušující vzorce, protože Control Web samotný maticové operace neumožňuje. Ve zdrojovém kódu jsou okomentované části operací. Pro výpočet dopravního zpoždění je tato metoda volána několikrát, postupně se posouvá pole u a podle chyby modelu se vyhodnocuje dopravní zpoždění.

6. Zhodnocení výsledků

V této kapitole představíme výsledky použitých metod identifikace. Dále zde budou zhodnoceny výsledky regulace pomocí PI a PID regulátoru nastavených podle představených metod. Simulace výsledků je provedena pomocí programu *Matlab*. Regulační pochod je poté modelován pomocí programu *Matlab-Simulink*. Identifikace probíhala ve vytvořeném modulu Control Webu na souboru nahraných reálných dat.

Na grafu 1 vidíme modře vykreslenou charakteristiku změřených dat. Tato charakteristika je odezvou na změnu vstupního napětí 0,1 V (červená charakteristika). Dále vidíme aproximovanou charakteristiku prvního řádu pomocí metody nejmenších čtverců, z grafu je patrné, že model prvního řádu dobře aproximuje naměřená data.



Graf 1: Naměřená a identifikovaná charakteristika č.1

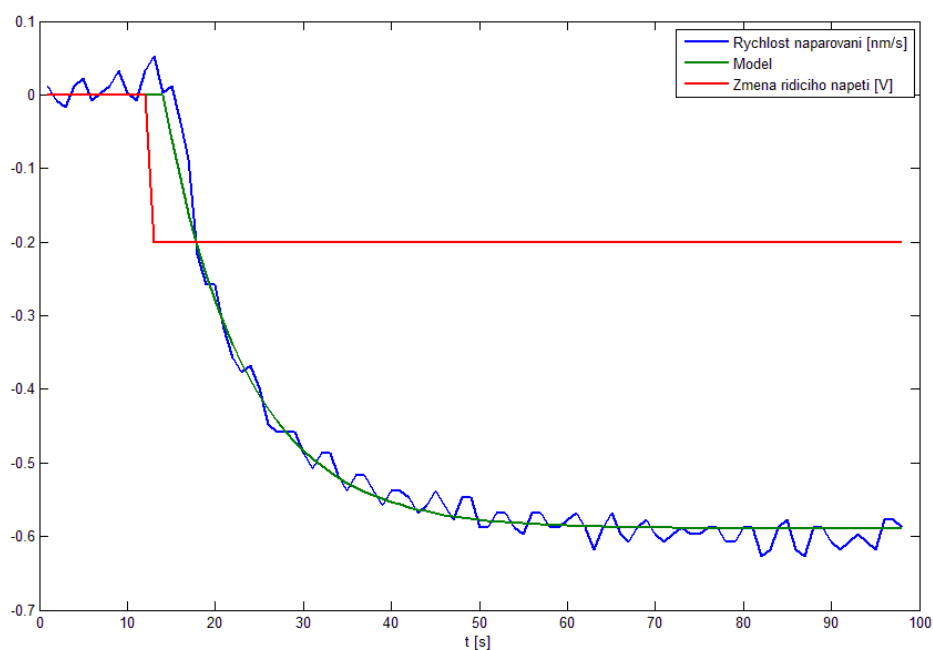
Data znázorněná v grafu 1 jsme aproximovali v modulu Control Web pomocí metody aproximace 1. řádem a metody nejmenších čtverců. Výsledky identifikace jsou v tabulce 1. Jak lze vidět, výsledky konstant jsou velice podobné. Rozdíly způsobuje

nejspíše pouze mediánový filtr a způsob zjištění časové konstanty u metody aproximace prvním řádem.

Tabulka 1: Tabulka porovnání hodnot použitých metod identifikace

	Zesílení k	Časová konstanta T_i [s]	Dopravní zpoždění θ [s]
Metoda nejmenších čtverců	6,65	12,25	2
Aproximace 1. řádem	6,67	12,76	2

V grafu 2 vidíme odezvu systému na záporný napěťový skok a její aproximaci metodou nejmenších čtverců.



Graf 2: Naměřená a identifikovaná charakteristika č.2

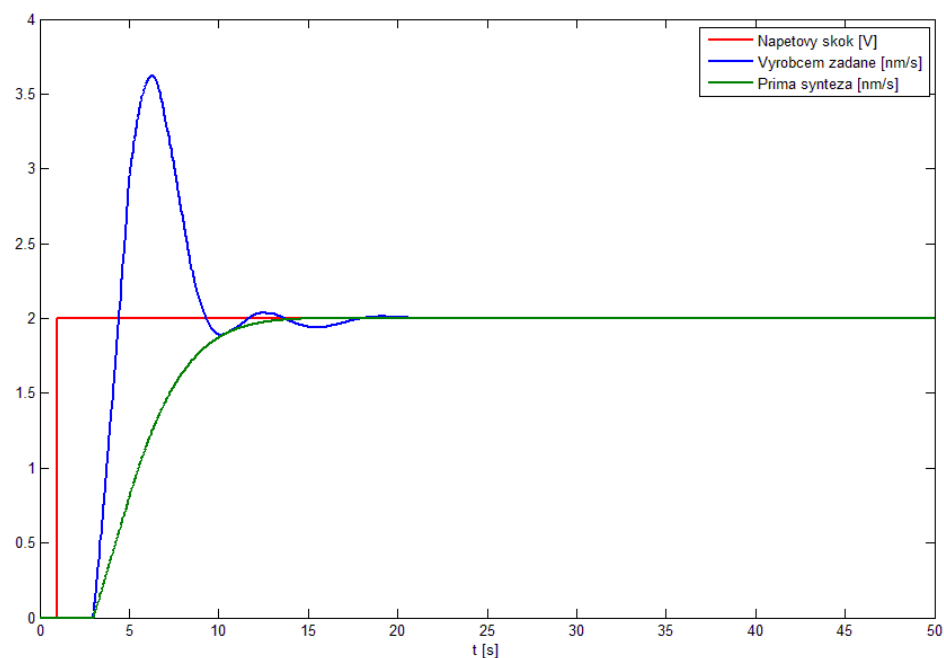
Zajímavostí na těchto datech je, že mají rozdílné zesílení a časovou konstantu proti datům, která jsou v grafu 1. To je způsobeno tím, že je soustava výparníku nelineární a

každá z přechodových charakteristik byla měřena v jiném pracovním bodě. V tabulce 2 jsou srovnány výsledky druhé identifikace.

Tabulka 2

	Zesílení k	Časová konstanta T_i [s]	Dopravní zpoždění θ [s]
Metoda nejmenších čtverců	2,95	9,35	2
Aproximace 1. řádem	3,04	9,20	2

V grafu 3 jsou vykresleny výstupní charakteristiky regulačního pochodu. Jsou zde srovnány výsledky dvou výše popsanych metod. Metoda zadaná výrobcem je agresivnější, spočátku překmitne žádanou hodnotu. Oproti přímé syntéze se ovšem dostane na žádanou hodnotu rychleji. Regulační pochod s PI regulátorem je pozvolnější, bez překmitu. Dynamika této metody se dá nastavit pomocí volitelného parametru τ_p .



Graf 3: Porovnání metod nastavení PID regulátoru

Závěr

V této bakalářské práci jsme se pokusili o realizaci programového modulu pro identifikaci napařovacího výparníku pro napařování tenkých vrstev. Z reálných dat jsme zjistili, že přechodovým charakteristikám velice dobře odpovídá model prvního řádu. Jako metody identifikace jsme tedy zvolili aproximaci prvním řádem s dopravním zpožděním a metodu nejmenších čtverců. Oběma metodami jsme dosáhli podobných výsledků, nutno však říct, že metoda nejmenších čtverců je mnohem robustnější a odolnější proti šumu. V programovém modulu tedy využíváme pro identifikaci metodou nejmenších čtverců. Z identifikovaných dat jsme navrhli konstanty PID regulátoru na základě dvou metod. První byla námi vybraná metoda přímé syntézy a druhá byla zadaná výrobcem. Každá metoda dává jiné výsledky, obecně ovšem nelze říct, která je pro regulaci odporového výparníku vhodnější. Volba sady konstant PID regulátoru je na volbě obsluhy napařovačky. Má tedy na výběr agresivnější či pozvolnější chod regulace.

Programový modul jsme naprogramovali a odzkoušeli na modelovém ovladači. V reálném prostředí nemohl být vyzkoušen, protože momentálně není funkční karta pro napařování SQM242.

Seznam literatury

- Fajmon, B., Růžicková, I.: Matematika 3. Skriptum FEKT VUT v elektronické formě, Brno 2013
- PZP. VNA 600/2: Vakuová napařovací aparatura. Dobruška, 2004.
- *Control Web 6 – dokumentace. Moravské přístroje a.s.* [online]. 2013 [cit. 2014-05-04]. Dostupné z: <http://www.mii.cz/cat?id=98&lang=405>
- NOSKIEVIČ, P.: Modelování a identifikace systémů. MONTANEX a.s., 1999, ISBN 80-7225-030-02
- VROŽINA, Milan, Zora JANČÍKOVÁ a Jiří DAVID. TECHNICKÁ UNIVERZITA OSTRAVA. *Identifikace systémů*. Ostrava, 2012. ISBN 978-80-248-2594-6
- KLÁN, Petr. O přímé syntéze PID regulátorů. Automa. 2006, č. 11.
- SQM-242 Thin Film Deposition Controller Card. INFICON [online]. 2013 [cit. 2014-05-05]. Dostupné z: <http://products.inficon.com/en-us/navigation/products/Product/Detail/SQM-242?path=Products%2FThinFilmDeposition%2F>
- Tyristorové regulační jednotky ELFIS[®] 1F. ELFIS [online]. [cit. 2014-05-04]. Dostupné z: http://www.elfis.cz/website/download/pdf_elfistory/1fx/_downloadFile.php